

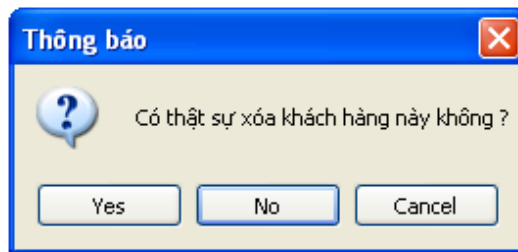
## BÀI 3

### KHAI BÁO VÀ SỬ DỤNG HỘP THOẠI

#### 1. Khai báo và sử dụng MessageBox

##### Lý thuyết

- Lớp MessageBox nằm trong namespace System.Windows.Forms và dẫn xuất từ lớp Object.
- Một đối tượng thuộc lớp MessageBox được dùng để trình bày thông báo bao gồm nội dung thông báo và các nút lệnh (OK, Yes, No, Cancel, ...) và chờ xác nhận từ phía người dùng.
- Lớp này chỉ có các phương thức (không có các loại thành viên khác), nhưng chủ yếu là các phương thức tĩnh Show để tạo và trình bày hộp thoại.





*Hình 3.1: Ví dụ về hộp thoại*

- Một số khai báo của phương thức Show:
  - ***public static DialogResult Show (string text)***
  - ***public static DialogResult Show (string text, string caption)***
  - ***public static DialogResult Show (string text, string caption, MessageBoxButtons buttons)***
  - ***public static DialogResult Show (string text, string caption, MessageBoxButtons buttons, MessageBoxIcon icon)***
  - ***public static DialogResult Show (string text, string caption, MessageBoxButtons buttons, MessageBoxIcon icon, MessageBoxDefaultButton defaultButton)***
- Ý nghĩa các tham số:
  - Tham số ***text***: nội dung của hộp thoại
  - Tham số ***caption***: tiêu đề của hộp thoại
  - Tham số ***buttons***: Quy định các nút lệnh thể hiện trên hộp thoại, nhận 1 trong các giá trị của ***enum MessageBoxButtons***, bao gồm:
    - ✓ ***OK***: chỉ có nút OK. Mặc định sử dụng hàm Show không có tham số buttons thì chỉ có nút OK
    - ✓ ***OKCancel***: 2 nút: OK và Cancel

- ✓ **AbortRetryIgnore**: 3 nút: Abort, Retry và Ignore
- ✓ **YesNoCancel**: 3 nút: Yes, No và Cancel
- ✓ **YesNo**: 2 nút: Yes và No
- ✓ **RetryCancel**: 2 nút Retry và Cancel
- Tham số **icon**: Quy định biểu tượng thể hiện trên hộp thoại, nhận 1 trong các giá trị của **enum MessageBoxIcon**, bao gồm:

Bảng 3.1: Các biểu tượng của hộp thoại

Giá trị	<i>None</i>	<i>Asterist</i>	<i>rror</i>	<i>Information</i>	<i>Question</i>
Biểu tượng	Không có			Như Asterist	
Giá trị	<i>Exclamation</i>	<i>Stop</i>	<i>Hand</i>	<i>Warning</i>	
Biểu tượng		Như Error	Như Error	Như Exclamation	

- Tham số **defaultButton**: Quy định nút lệnh được chọn mặc định khi nhấn Enter, nhận 1 trong các giá trị của **enum MessageBoxDefaultButton**, bao gồm: Button1, Button2, Button3; mặc định là Button1

## Thực hành

- Nếu chỉ tạo hộp thoại để đưa ra thông báo thì chỉ cần gọi phương thức Show và truyền các tham số cần thiết như ví dụ sau:

```
MessageBox.Show("Đã xóa khách hàng vừa chọn!")
```

- Nếu cần lấy về nút xác nhận từ phía người dùng để xử lý chương trình theo các hướng khác nhau thì cần khai báo biến để lấy giá trị xác nhận, như ví dụ sau:

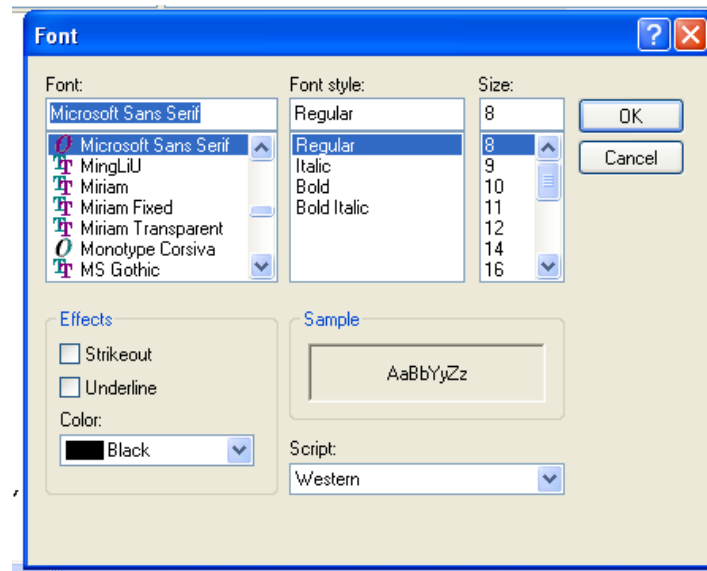
```
DialogResult traloi = MessageBox.Show("Có thật sự xóa
khách hàng này không ?", "Thông báo",
MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question);
if (traloi == DialogResult.Yes)
    MessageBox.Show("Đã xóa!");
else if (traloi == DialogResult.No)
    MessageBox.Show("Không xóa!");
else
    MessageBox.Show("Không làm gì cả!");
```

## 2. Khai báo và sử dụng hộp thoại Font

### Lý thuyết

- Giới thiệu về lớp CommonDialog:
  - CommonDialog là lớp trừu tượng định nghĩa các thành viên để tạo và hiển thị các hộp thoại thông thường của Windows.

- Phương thức ***public DialogResult ShowDialog()***: hiển thị hộp thoại, chờ xác nhận từ người dùng và trả về 1 trị của enum DialogResult tùy thuộc vào nút được nhấn.
  - Các lớp dẫn xuất từ lớp CommonDialog bao gồm: FontDialog, ColorDialog.
- Giới thiệu về lớp FontDialog:
- Đối tượng thuộc lớp FontDialog dùng để trình bày hộp font và cho phép chọn 1 font chữ.

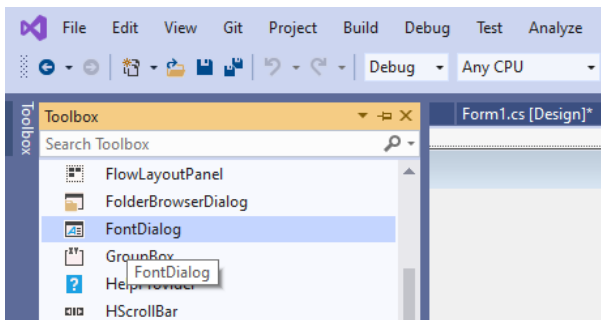


Hình 3.2: Ví dụ về FontDialog

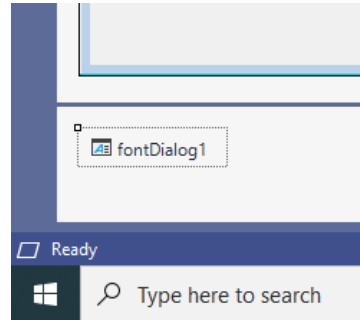
- Một số thuộc tính của lớp FontDialog:
  - ✓ ***public Font Font { get; set; }***: gán hoặc lấy về đối tượng font được chọn trong hộp thoại.
  - ✓ ***public bool ShowColor { get; set; }***: có hiển thị màu chữ trong hộp thoại font không.
  - ✓ ***public Color Color { get; set; }***: gán hoặc lấy về màu chữ được chọn trong hộp thoại.

## Thực hành

- Bước 1: Tạo đối tượng thuộc lớp FontDialog
  - Cách 1: Kéo thả đối tượng FontDialog từ Toolbox qua form



Hình 3.3: Đối tượng FontDialog trên thanh Toolbox



Hình 3.4: Đối tượng FontDialog trên form

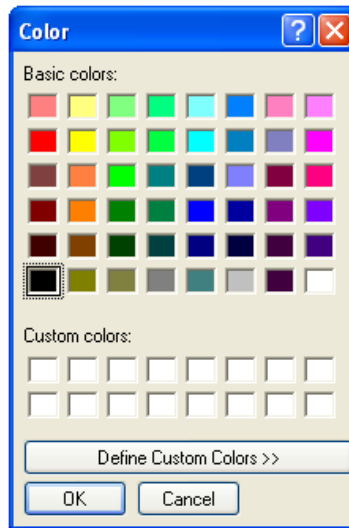
- Cách 2: Sử dụng hàm tạo để tạo đối tượng FontDialog  
`FontDialog ftd = new FontDialog();`
- Bước 2: Khai báo các thuộc tính khác (nếu cần).
- Bước 3: Gọi phương thức ShowDialog để mở hộp thoại  
`ftd.ShowDialog();`
- Bước 4: Lấy các giá trị font và màu được chọn trên hộp thoại để gán cho các thuộc tính khác. (Ví dụ như gán cho font chữ và màu chữ của button1 sau đây).  
`Font ff = ftd.Font;`  
`Color cr = ftd.Font;`  
`button1.Font = ff;`  
`button1.ForeColor = cr;`

## 3. Khai báo và sử dụng hộp thoại Color

### Lý thuyết

Giới thiệu về lớp ColorDialog:

- Đối tượng thuộc lớp ColorDialog dùng để trình bày hộp màu và cho phép chọn 1 màu.



Hình 3.5: Ví dụ về ColorDialog

- Thuộc tính Color:

***public Color Color { get; set; }:*** gán hoặc lấy về màu được chọn trong hộp thoại.

### Thực hành

Thực hiện tương tự như đối với hộp thoại Font

## 4. Khai báo và sử dụng hộp thoại Open File

### Lý thuyết

- Giới thiệu về lớp OpenFileDialog:
  - OpenFileDialog là lớp trừu tượng định nghĩa các thuộc tính và phương thức cho các lớp dẫn xuất. Lớp OpenFileDialog có 2 lớp dẫn xuất là OpenFileDialog và SaveFileDialog.
  - Một số thuộc tính thường dùng:
    - ✓ ***public string Title { get; set; }:*** tiêu đề hộp thoại
    - ✓ ***public string Filter { get; set; }:*** xác định phần mở rộng tập tin mà hộp thoại hiển thị

#### Ví dụ:

```
ofd.Filter = "Text (*.txt)|*.txt|Picture (*.bmp;*.ico)|*.bmp; *.ico";
```

- ✓ ***public string InitialDirectory { get; set; }:*** đường dẫn mặc định khi vừa mở hộp thoại

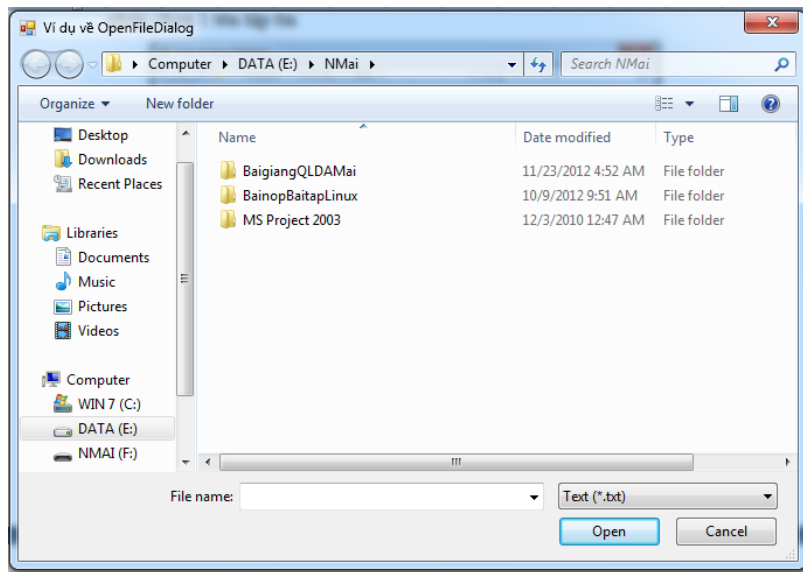
#### Ví dụ:

```
ofd.InitialDirectory = @"D:\Baitap";
```

- ✓ ***public string FileName { get; set; }:*** tên tập tin được chọn, bao gồm cả đường dẫn và phần mở rộng
- ✓ ***public string[] FileNames { get; }:*** tên tất cả các tập tin được chọn.

- Giới thiệu về lớp OpenFileDialog:

- Đối tượng thuộc lớp OpenFileDialog dùng để trình bày hộp thoại Open cho phép chọn 1 hoặc nhiều tên tập tin.



Hình 3.6: Ví dụ về OpenFileDialog

- Ngoài các thuộc tính của lớp FileDialog, lớp OpenFileDialog còn có thêm các thuộc tính, trong đó có thuộc tính MultiSelect:

***public bool Multiselect { get; set; };*** cho phép chọn nhiều file

### Thực hành

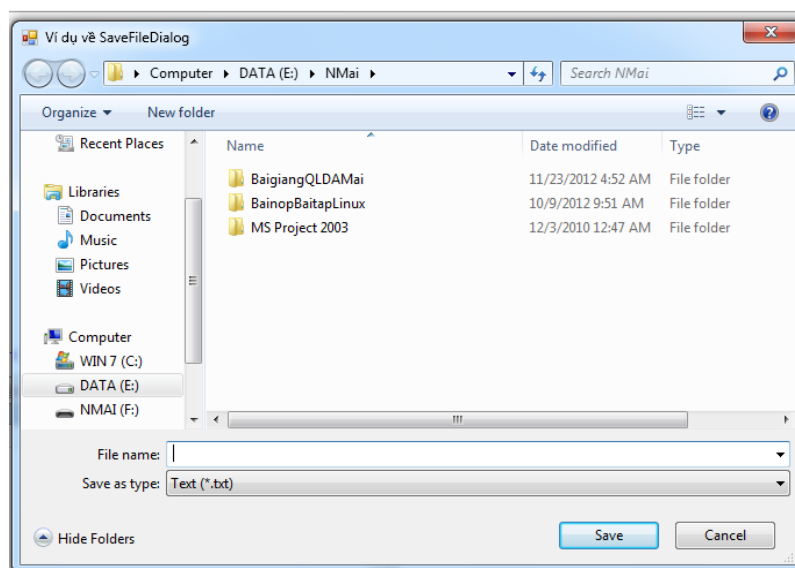
Thực hiện tương tự như đối với hộp thoại Font

## 5. Khai báo và sử dụng hộp thoại Save File

### Lý thuyết

Giới thiệu về lớp SaveFileDialog:

- Đối tượng thuộc lớp SaveFileDialog dùng để trình bày hộp thoại Save As cho phép nhập hoặc chọn 1 tên tập tin (nhằm lưu trữ)



Hình 3.7: Ví dụ về SaveFileDialog

- Ngoài các thuộc tính của lớp `FileDialog`, lớp `SaveFileDialog` còn có thêm các thuộc tính, trong đó có thuộc tính `OverwritePrompt`:

**`public bool OverwritePrompt { get; set; }`**: hiển thị hộp thoại cảnh báo khi chuẩn bị overwrite 1 tập tin đã tồn tại.

## Thực hành

Thực hiện tương tự như đối với hộp thoại `Font`.

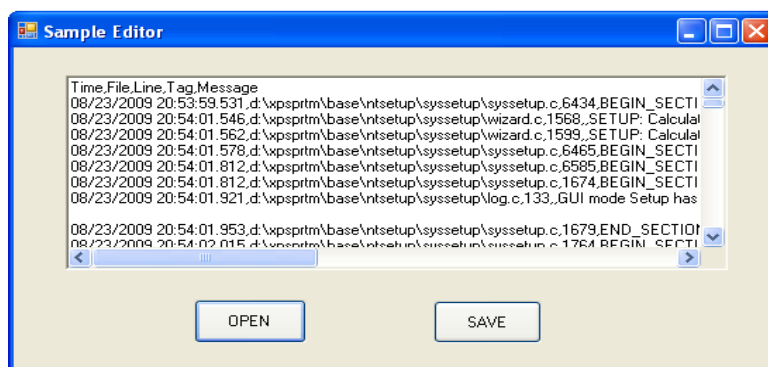
## Câu hỏi và bài tập

### 1. Tạo form theo yêu cầu sau:

- Khi click chuột lên nền form thì mở hộp thoại màu
- Nếu có 1 màu được chọn và nhấn nút `OK` thì kiểm tra xem đó có phải là 1 màu có tên sẵn không, nếu có thì in tên màu, ngược lại in các giá trị `ARGB`
- Tạo 1 nút lệnh `Font` cho phép mở hộp thoại font và chọn 1 font làm font chữ cho nút lệnh

### 2. Tạo form như hình:

- Nút `OPEN` hiện hộp thoại `Open` cho phép chọn 1 tập tin `txt` trên đĩa và nạp nội dung vào `RichTextBox`
- Nút `SAVE` hiện hộp thoại `Save As` cho phép chọn tên tập tin cũ hoặc nhập 1 tên mới để lưu nội dung trong `RichTextBox` ra tập tin `txt` trên đĩa
- Thêm 1 nút lệnh `List FileName` để mở hộp thoại `Open` và liệt kê tất cả các tên của các tập tin vừa chọn trong hộp `Open`
- Cho mã lệnh hàm `ReadFile` để đọc nội dung tập tin vào `RichTextBox` `rtb`:



```
using System.IO;

void ReadFile(string path)
{
    if (path != "")
    {
        StreamReader sr = new
        StreamReader(path);
        rtb.Text = sr.ReadToEnd();
        sr.Close();
    }
}
```

```
    }  
}
```

- Cho mã lệnh hàm WriteFile để ghi nội dung RichTextBox rtb vào tập tin:

```
void WriteFile(string fullpath)  
{  
    try  
    {  
        using (StreamWriter sw = new  
StreamWriter(fullpath))  
        {  
            sw.WriteLine(rtb.Text);  
            sw.Flush();  
            sw.Close();  
        }  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message);  
    }  
}
```